Congratulations! You passed!

TO PASS 80% or higher

Keep Learning

GRADE
100%

Test your Project understanding

LATEST SUBMISSION GRADE
100%

1. How many csv file does our dataset have ?

1 / 1 point

- ○ 1
- ○ 3
- ◉ 2

Correct
we have two csv files in our dataset:

1- listening.csv

2-genre.csv

2. In distributed data processing we divide the data into smaller parts called...

1 / 1 point

- ○ Partitions
- ◉ RDD
- ○ objects

Correct
In distributed data processing we divide the data into smaller parts called **RDD**

3. While loading a csv file why we put the option ***inferSchema*** to be true?

1 / 1 point

- ◉ this option will infer column types based on the dataset
- ○ this option will set the column names as they were mentioned in the csv file

Correct
TRUE! it automatically infers column types based on the data

4. While loading a csv file why we put the option ***header*** to be true?

1 / 1 point

- ○ this option will set the column names like _C0, _C1 and ...
- ◉ this option will set the column names as they were mentioned in the csv file

✓ **Correct**
Correct! this option will set the column names as they were mentioned in the csv file

5. which method is used for showing the content of a pyspark dataframe ? **1 / 1 point**

- ○ .print()
- ◉ .show()
- ○ .head()

✓ **Correct**
Correct!

6. how we can delete **rows with null values** in pyspark dataframe ? **1 / 1 point**

- ○ df.dropna()
- ◉ df.na.drop()
- ○ df.drop(Null)

✓ **Correct**
Incorrect!

df.na.drop() is used for deleting rows with null values

7. How to drop a column in pyspark dataframe ? **1 / 1 point**

- ◉ df.drop('column_name')
- ○ df.column_name.drop()
- ○ df.delet(column_name)

✓ **Correct**
correct!

df.drop('column_name')

8. how to select *col1* and *col2* from pyspark dataframe, *df* ? **1 / 1 point**

- ○ df[[col1,col2]]
- ◉ df.select('col1','col2')

✓ **Correct**

correct!

the correct answer is df.select('col1','col2')

9. which of the following methods is used as *where clause* in sql to select rows that satisfy a condition of a pyspark dataframe? **1 / 1 point**

- ○ df.where(*condition*)
- ◉ df.filter(condition)
- ○ df[df[condition]]

✓ **Correct**

correct!

df.filter(condition) is the correct answer

10. which option is the correct one for inner joining two pyspark dataframes df1 and df2 on column col? **1 / 1 point**

- ◉ df1.join(df2 ,on=['col'], how='inner')
- ○ df1.merge(df2 ,on=['col'], how='inner')

✓ **Correct**

correct!

to inner join df1 and df2 on column *col :*

df1.join(df2 ,on=['col'], how='inner')